

*The video link:*

<https://youtu.be/CMUYEB7QcBU>

# Physical Computing Project

Made by Yufan Liu,  
Xi Wang,  
Liaoran Zhang

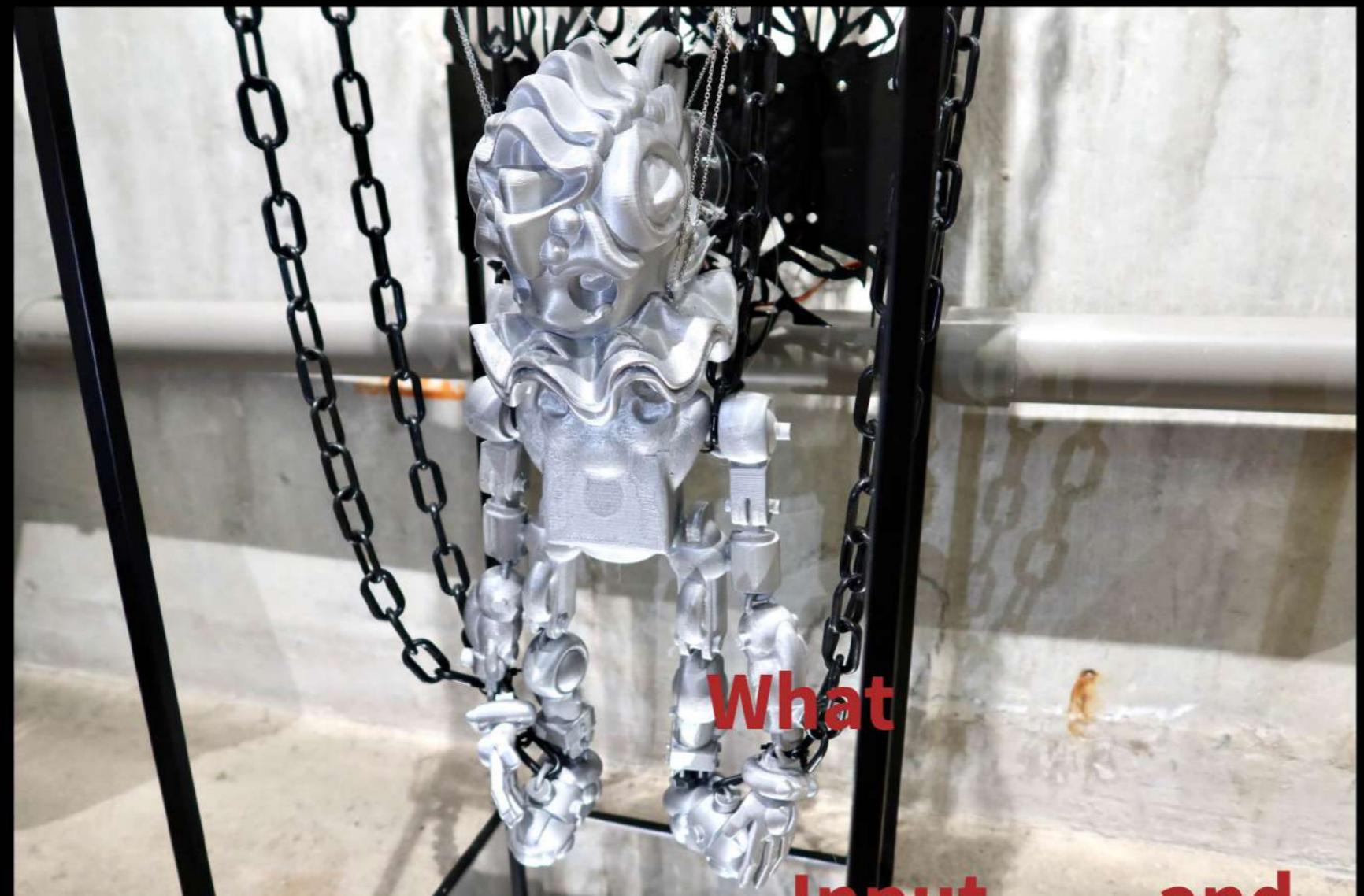


## “Flamenco”

make  
it?

How  
to

About  
our  
“Flamenco”



What  
Input  
and

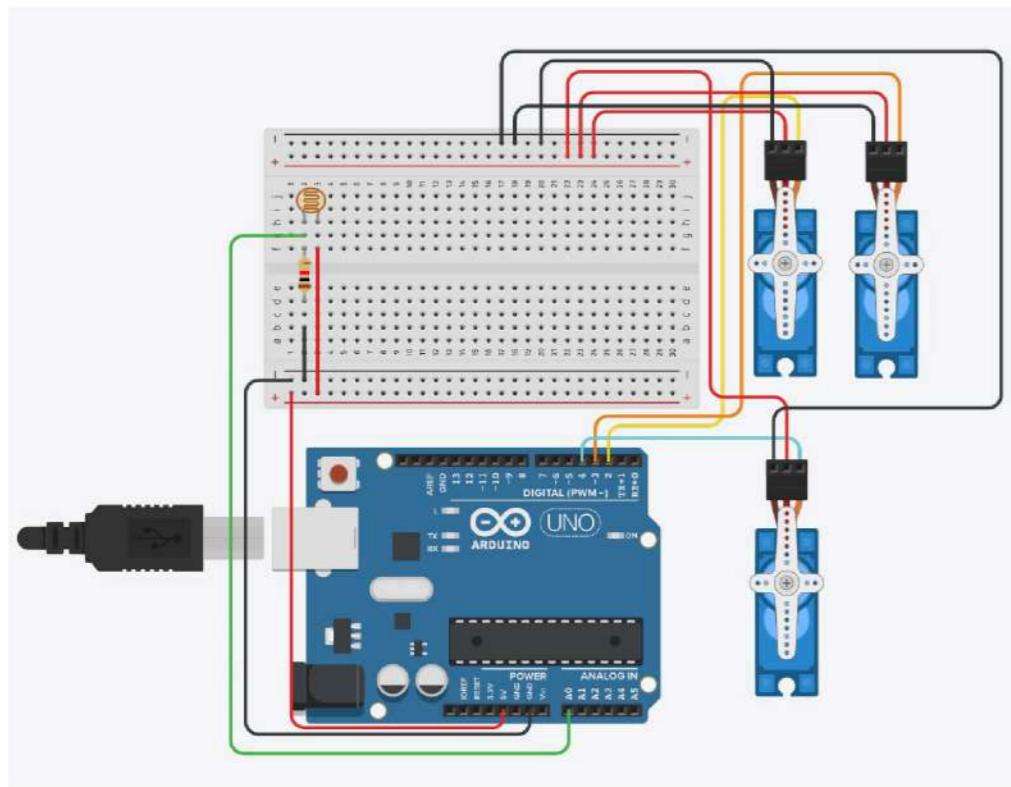
Output

Trivial emptiness, and how  
do you fight against the  
powerlessness and frus-  
tration of your whole life?

?  
we use

# Circuit of using light dependent sensor to control 3 servos

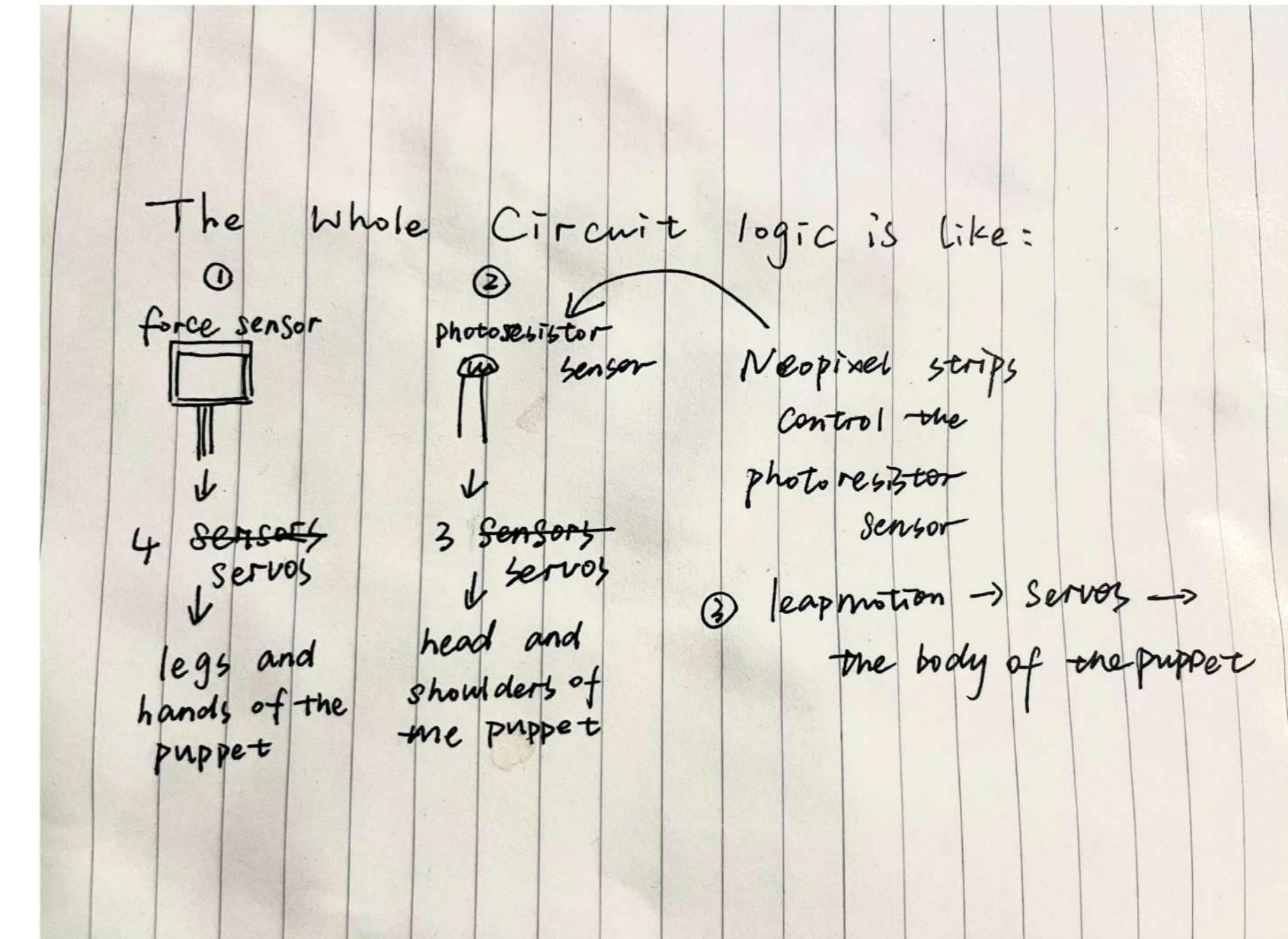
A testing of using light dependent sensor to control servos.



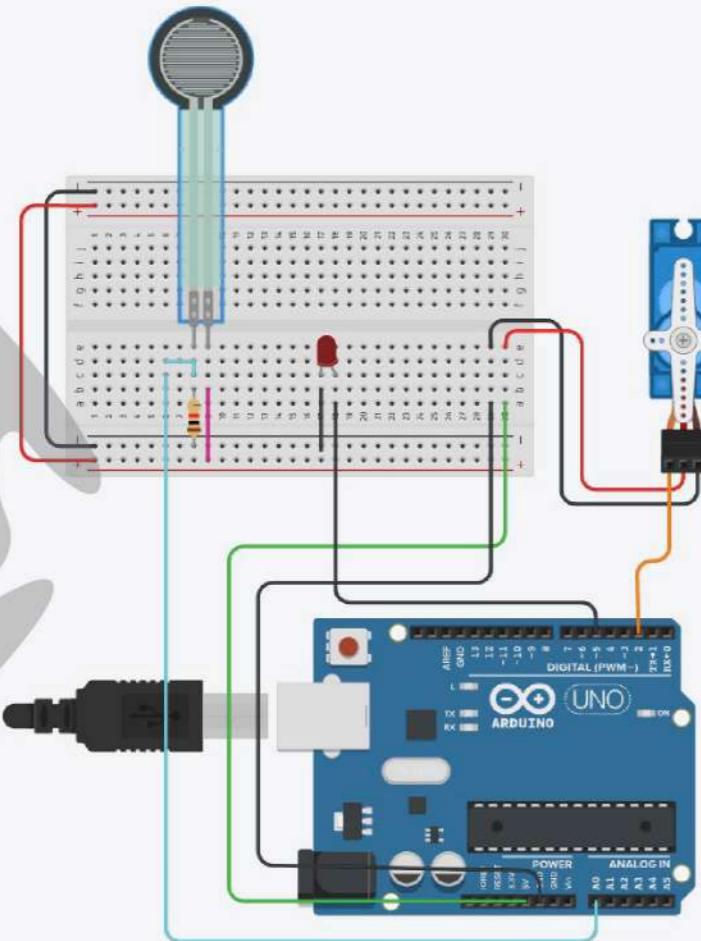
```
void loop() {
    lightval=analogRead(lightpin);
    Serial.println(lightval);
    delay(tm);

    angle= lightval/5;
    myservo1.write(angle);
    myservo2.write(angle);
    myservo3.write(angle);
    Serial.println("angle is");
    Serial.println(angle);

}
```



# Circuit of fsr and servos



A testing of using force sensor to control 4 servos.

At first I was going to put a few leds on this arduino board. But after this I had some new ideas about the leds.

```
void loop()
{
    fsr = analogRead(A0);

    if (fsr >= 0 && fsr < 50) {
        Serial.println(fsr); // getting know if the fsr go into this area
        for (pos = 0; pos <= 45; pos += 1) { // goes from 0 degrees to 45 degrees

            servo_1.write(pos); // tell servo to go to position in variable 'pos'
            servo_2.write(pos);
            servo_3.write(45+pos);
            servo_4.write(45+pos);
            delay(5); // waits 5 ms for the servo to reach the position
        }
        for (pos = 45; pos >= 0; pos -= 1) { // goes from 45 degrees to 0 degrees

            servo_1.write(pos); // tell servo to go to position in variable 'pos'
            servo_2.write(pos);
            servo_3.write(pos-45);
            servo_4.write(pos-45);
            delay(5); // waits 5 ms for the servo to reach the position
        }
    }

    else if (fsr >= 50 && fsr <= 300){
        Serial.println(fsr);

        for (pos = 0; pos <= 90; pos += 1) { // goes from 0 degrees to 90 degrees

            servo_1.write(pos); // tell servo to go to position in variable 'pos'
            servo_2.write(pos);
            servo_3.write(pos+45);
            servo_4.write(pos+45);
            delay(5); // waits 5 ms for the servo to reach the position
        }
        for (pos = 90; pos >= 0; pos -= 1) { // goes from 90 degrees to 0 degrees
    }
```

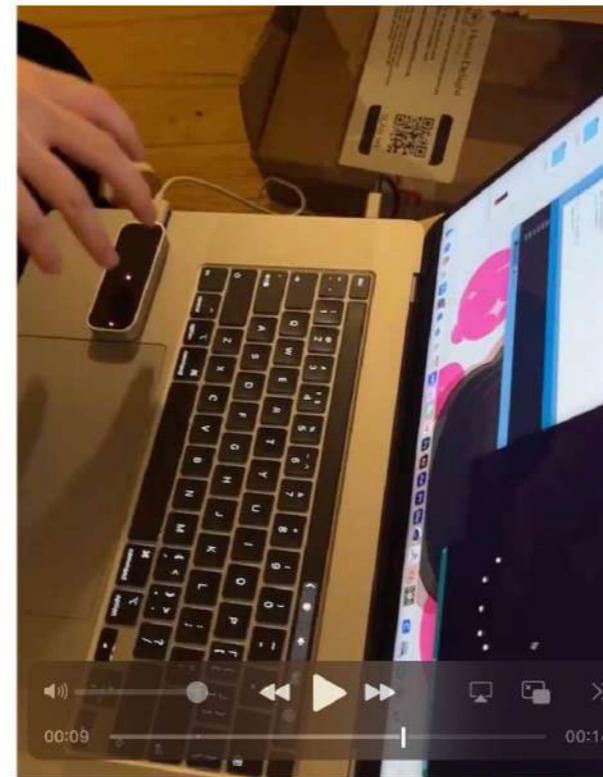
# Leapmotion control servos

## The code in processing:

```
// set com port. Currently: "/dev/tty.usbmodemfd121"
println("Available serial ports:");
printArray(Serial.list());
port = new Serial(this,Serial.list()[3], 9600);
}

public void draw() {
background(0);
fill(255);
for (Finger f : leap.getFingerList()) {
PVector position = leap.getTip(f);
//PVector velocity = leap.getVelocity(f);
ellipse(position.x, position.y, 10, 10);
if (position.y > 720) {
angle = 180;
} else if (position.y < 0) {
angle = 0;
} else {
angle = int((position.y) / 4);
}
port.write(angle);
println(angle);
}
}

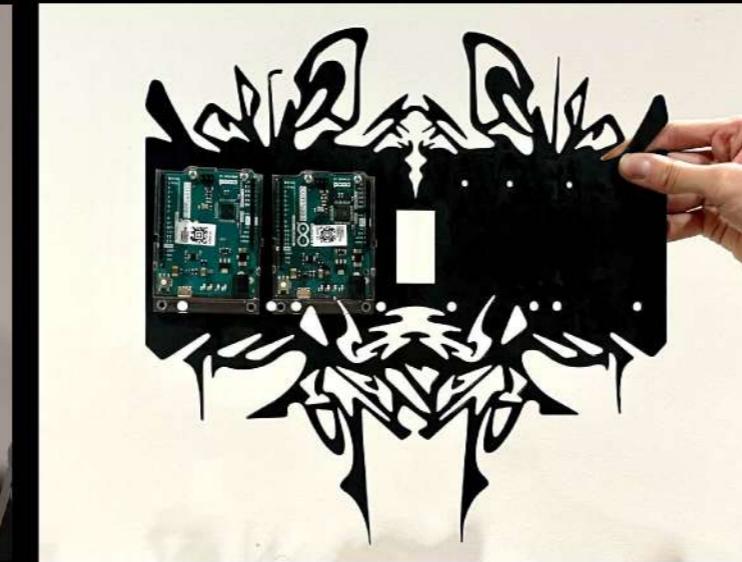
public void stop() {
leap.stop();
}
```



## The code in Arduino:

```
#include <Servo.h>
Servo myServo;
//int handPos;
//int angle;
void setup() {
myServo.attach(6);
Serial.begin(9600);
myServo.write(0);
}
void loop() {
byte angle;

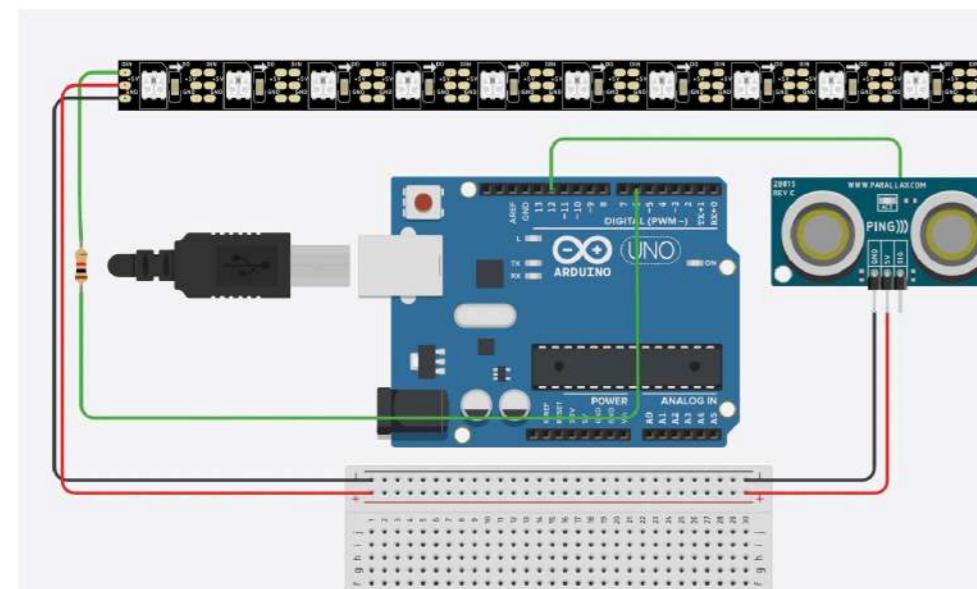
if (Serial.available()) {
// Read angle from Processing
angle = Serial.read();
Serial.println(angle);
// If fingers in window, read servo angle
myServo.write(angle);
}
}
```



This is a picture of the draft model of the structure of the light and the fixed arduino board to the molding process



# Neopixel Strips



The TrigPin is 12 and Echo is 13.  
These are screenshots of key parts of the code,  
The main logic of the code is to use the distance sensor  
to control the light strip changes in sections.

```
void NeoBlink(int num, int wait)
{
    if(distance <20)
    {Serial.print("00");
        for (int i = 0; i < num; i++)
        {
            pixels1.setPixelColor(i, 14, 221, 162);
            pixels2.setPixelColor(i, 14, 221, 162);
        }
        pixels1.show();
        pixels2.show();

    }else if(distance>20 && distance<40){
        Serial.print("22");
        for (int j = 0; j < num; j++)
        {
            pixels1.setPixelColor(j, 255, 77, 230);
            pixels2.setPixelColor(j, 255, 77, 230);
        }
        pixels1.show();
        pixels2.show();
        delay(10);
    }
}
```

```
void setup()
{
    pinMode(trigPin , OUTPUT);
    pinMode(echoPin , INPUT);
    pixels1.begin();
    pixels2.begin();
    pixels1.setBrightness(50);
    pixels2.setBrightness(50);
    Serial.begin(9600);
}

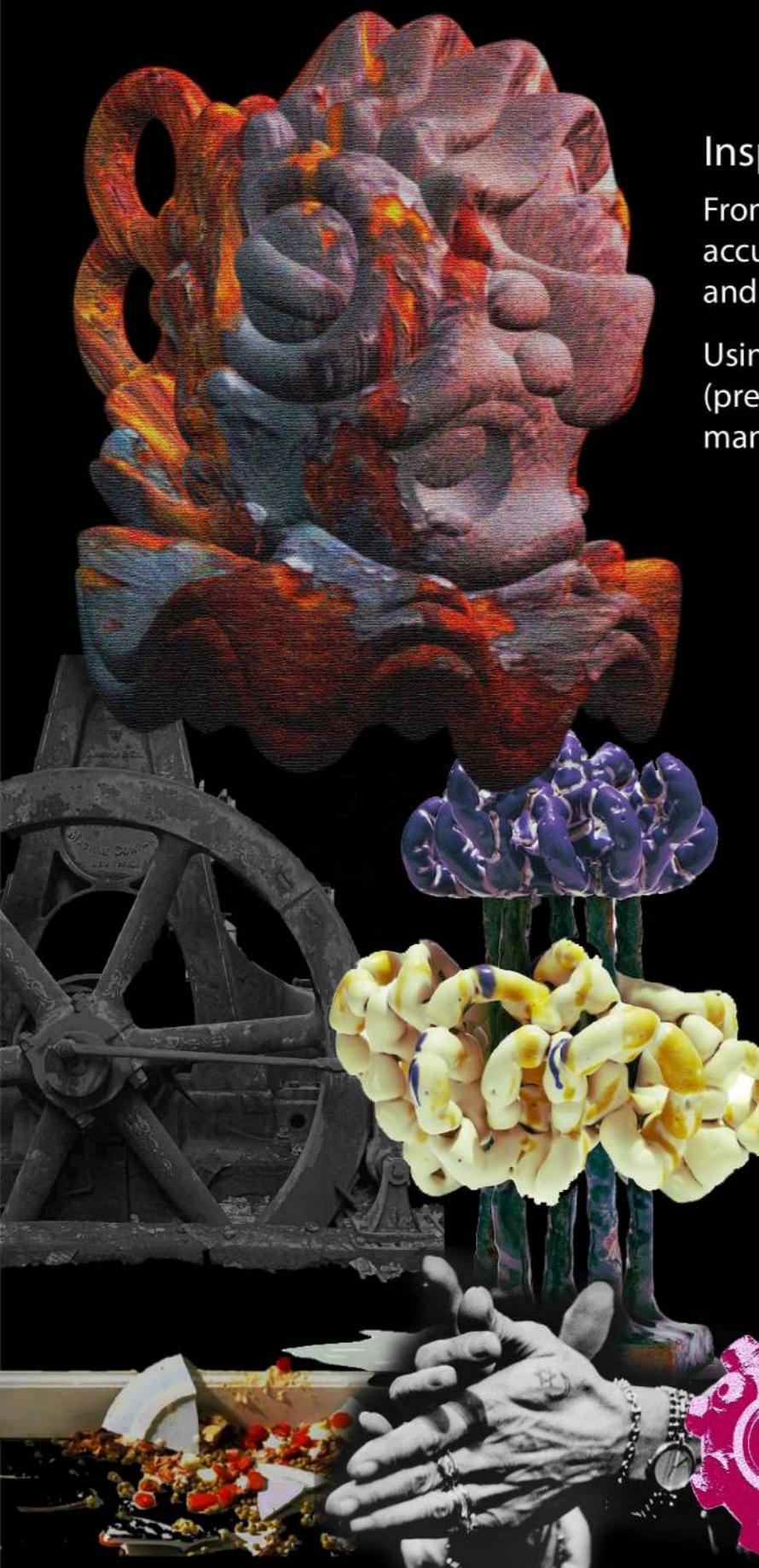
void loop()
{
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    distance = duration/58.2;
    NeoBlink(19, 500);

    delay(100);
}
```

## Assembly and fixing



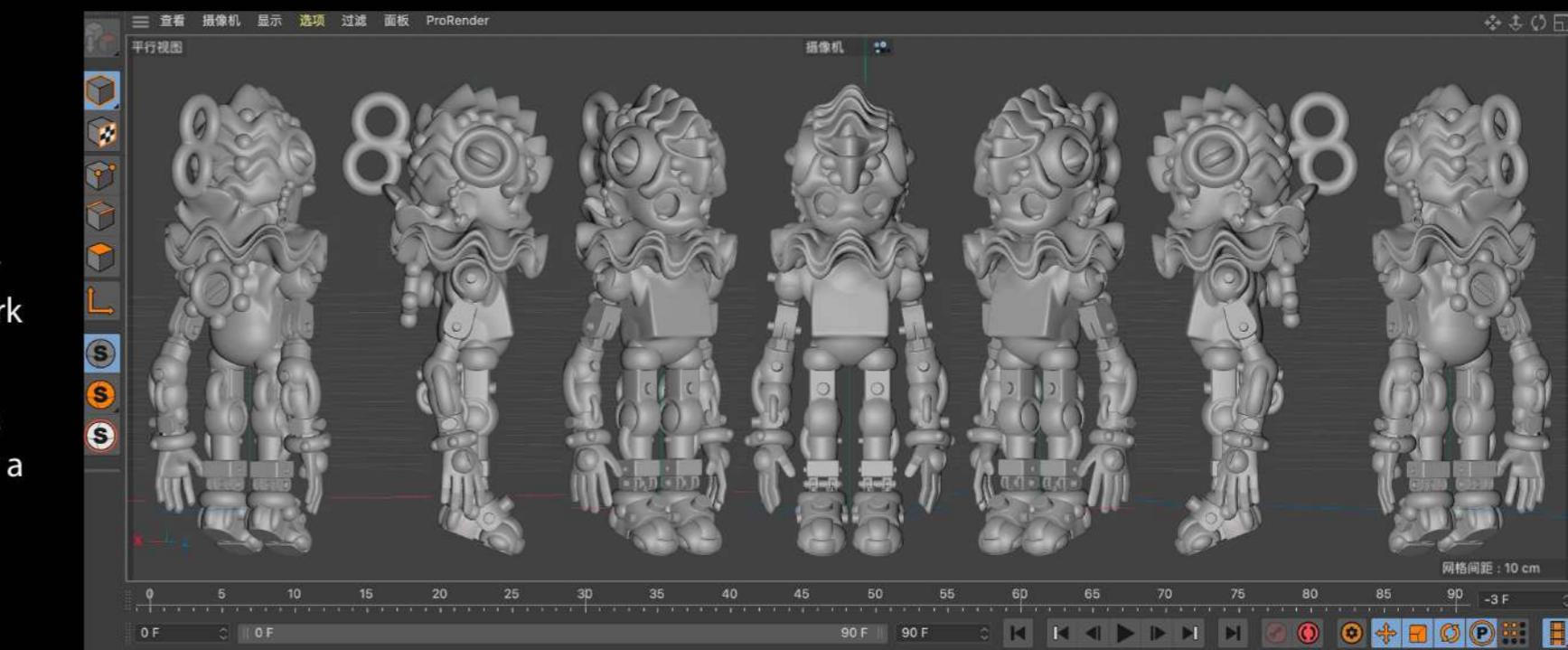
# Model



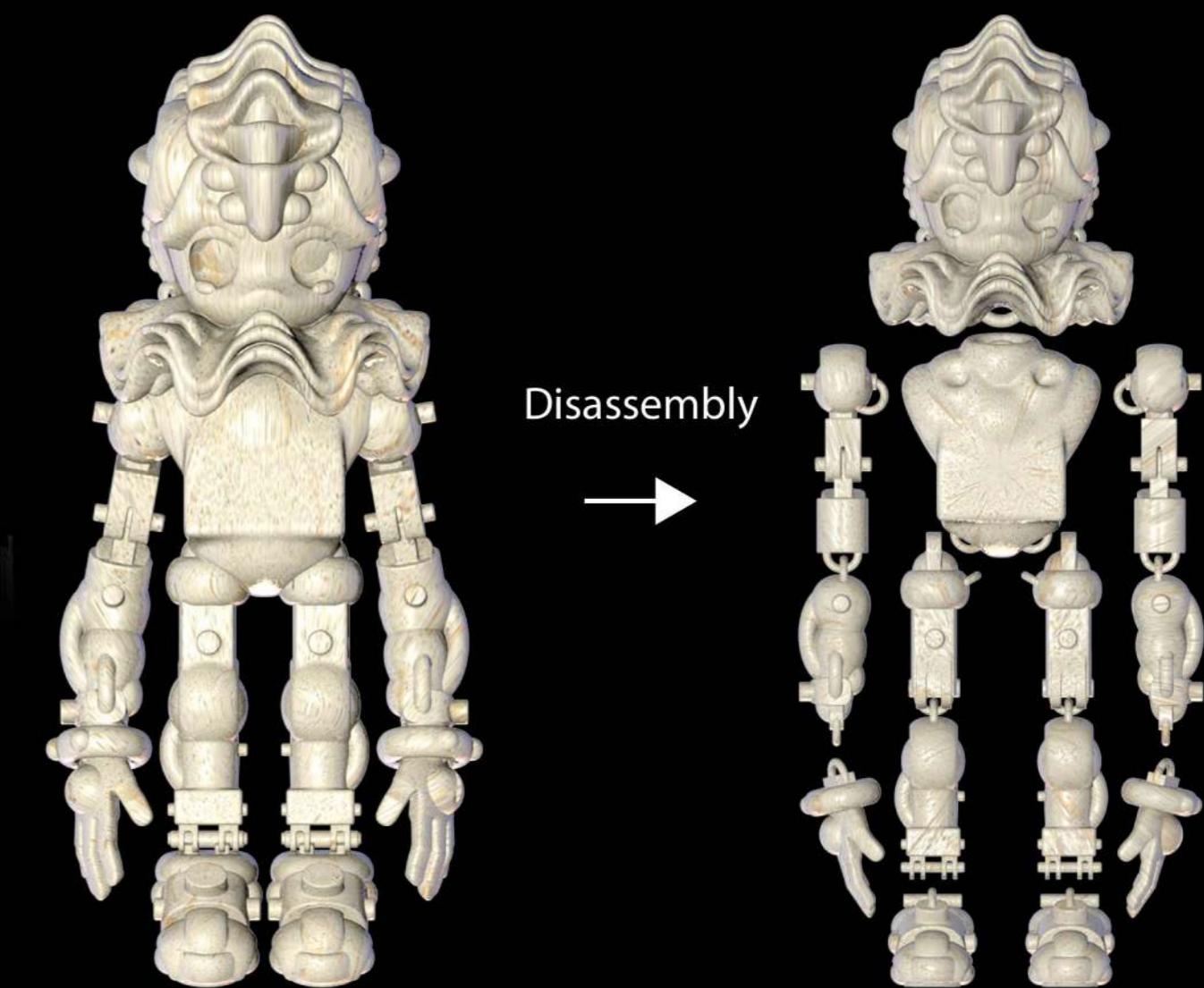
## Inspiration

From the starting point of "fatalism" and "life under accusation", the elements of the valve, the clockwork and the goldfish in the bag are chosen.

Using a mannequin manipulated by invisible forces (predestination, capital, values ..... ) manipulated by a mannequin to express the theme

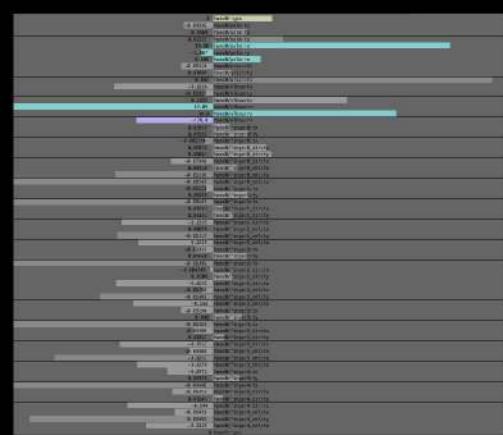


Disassembly

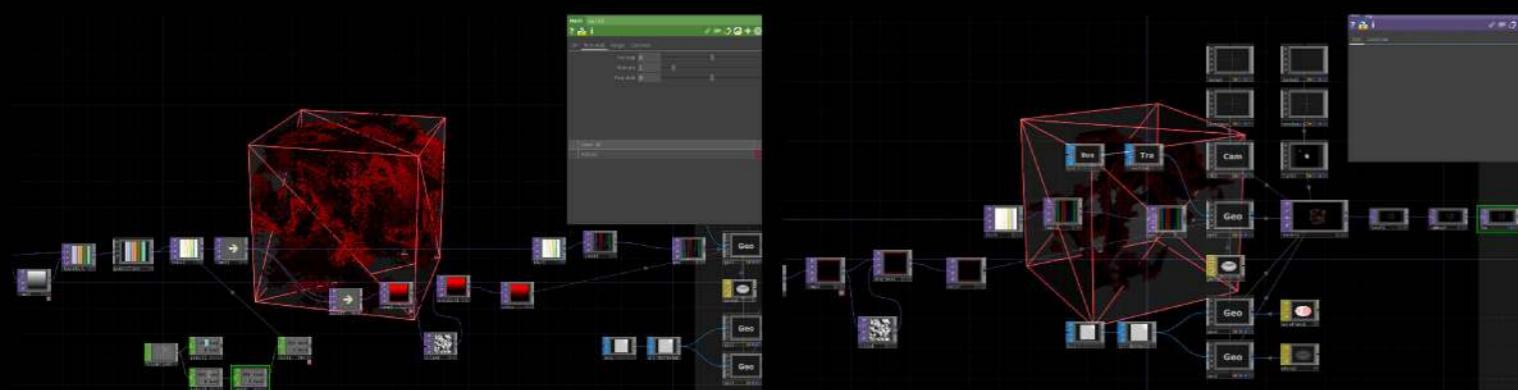
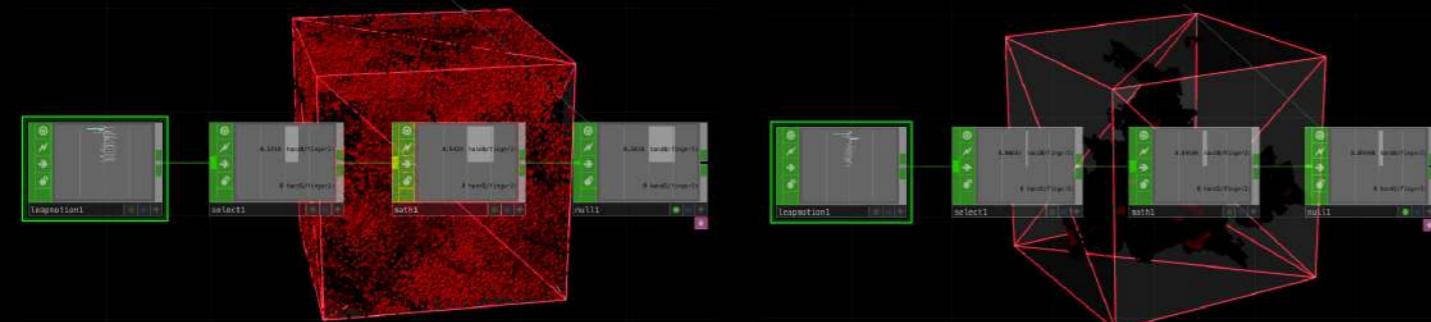
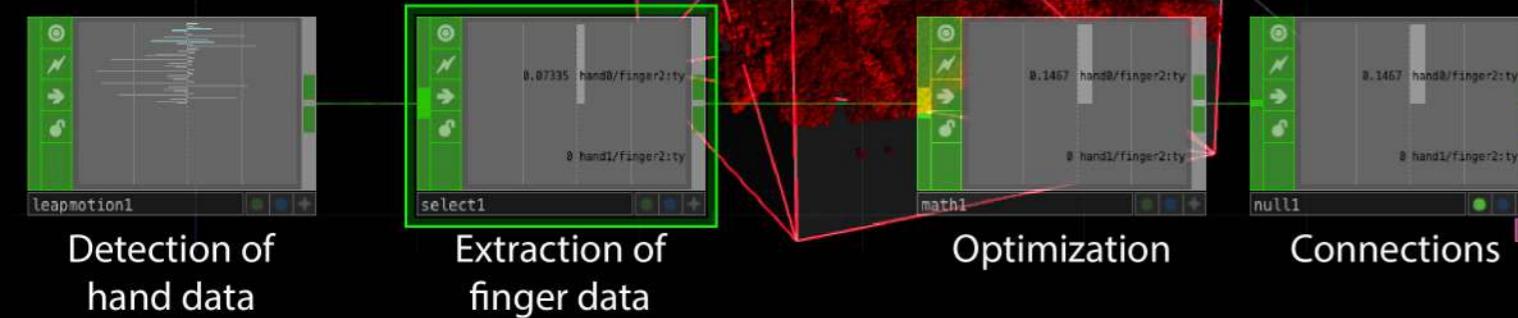


# Leap motion + TD

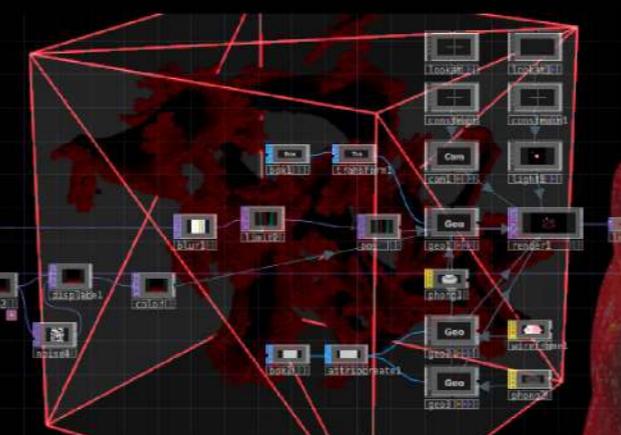
## Leapmotion



Cue invisible pressure by  
Leapmotion recognition of  
rising and falling hand  
gestures



## Touchdesigner



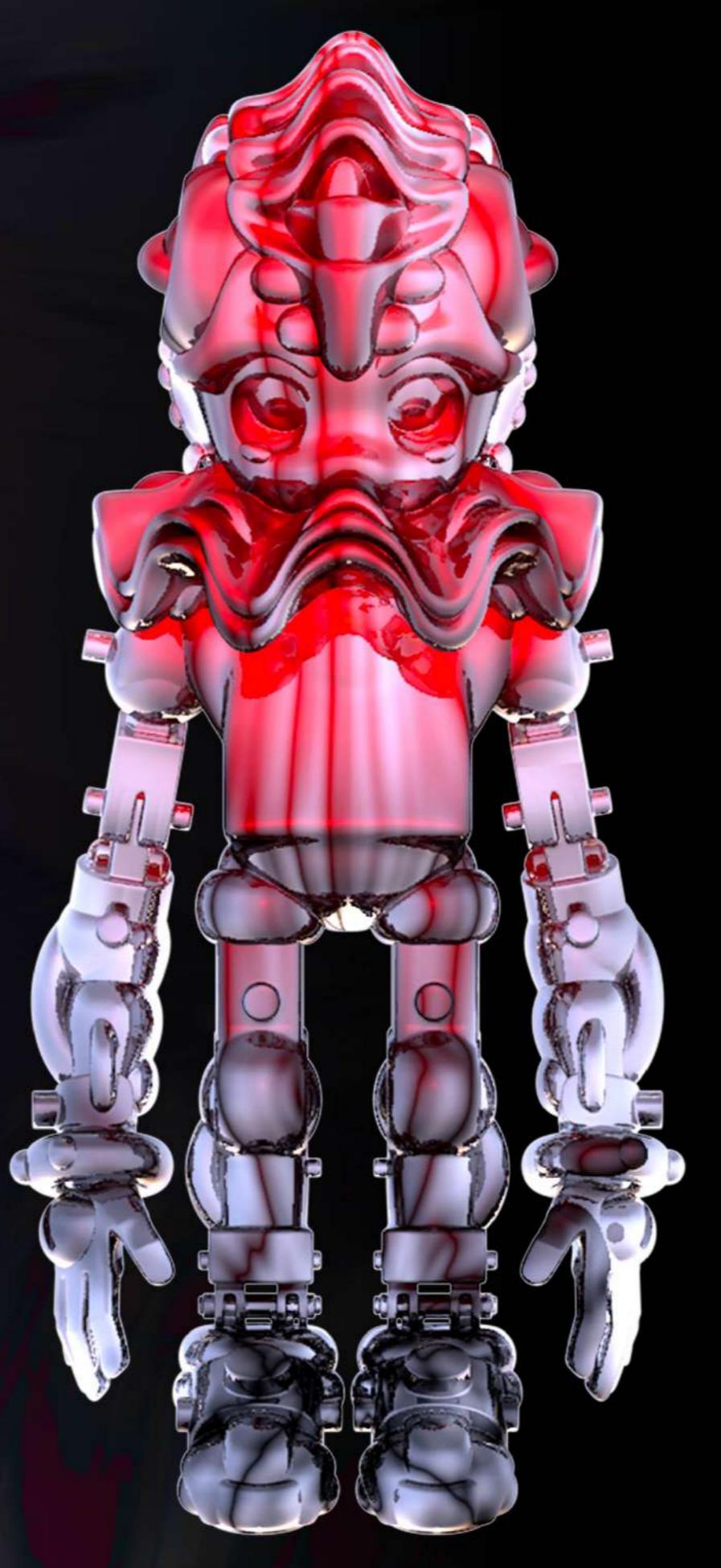
- Gesture down particle graphics shrink and appear
- Particle pattern expansion when the gesture rises



## Effect

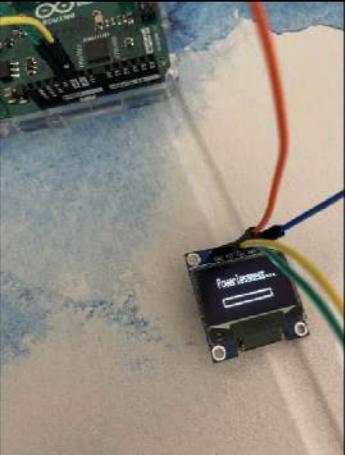
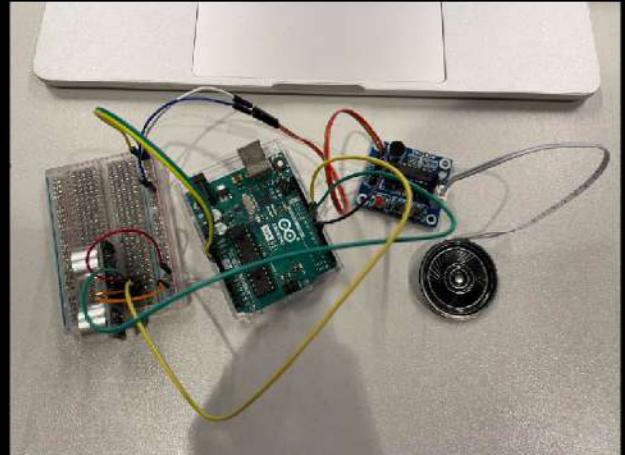


Model rendering

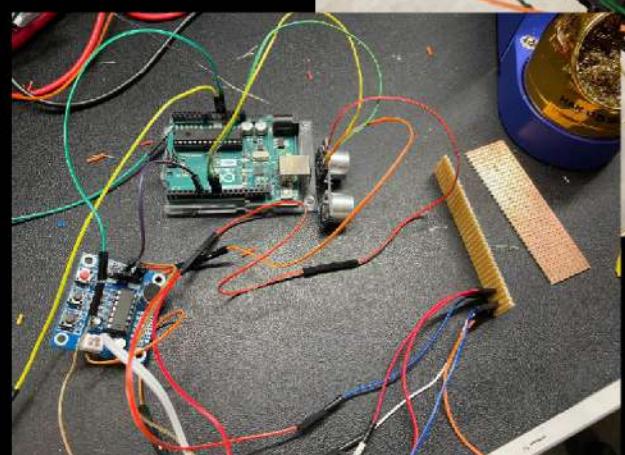


# Sound & OLED

## Process



The installation emits a sound when one approaches the ultrasonic sensor. The original sound I set up was a crashing sound to express the feeling of repeatedly fighting against powerlessness and frustration. People can also record their own voice by pressing the red button. The installation will emit the recorded sound.



The OLED screen shows the name of our work and a progress bar that is permanently at 0%. This indicates a feeling of repeated frustration and the inability to change the final 0% result.

## Code

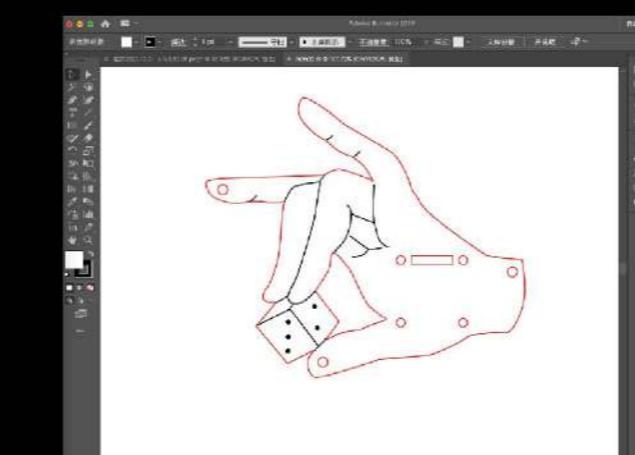
Below are screenshots of the circuit diagram and code.

```
void Draw_Init_Interface(void) {
    for (size_t i = 0; i < 46; i = i + 5)
    {
        display.clearDisplay();
        display.setTextSize(1);
        display.setTextColor(WHITE);
        display.setCursor(38, 8);
        display.println("FLAMINGO");
        display.setCursor(100, 20);
        display.println("0%");
        display.drawRect(30, 20, 65, 6, WHITE);
        display.drawLine(40, 40, 40 + i, 40, WHITE);
        display.drawLine(40, 41, 40 + i, 41, WHITE);
        display.display();
        //delay(10);
        display.clearDisplay();
    }
    display.display();
}
void loop() {
    Draw_Init_Interface();
}
```

```
#define trigPin 9
#define echoPin 8
#define voice 6
int sound = 250;
void setup() {
    Serial.begin(9600);
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    pinMode(voice, OUTPUT);
}
void loop() {
    Serial.begin(9600);
    long duration, distance;
    digitalWrite(trigPin, LOW);
    delay(2);
    digitalWrite(trigPin, HIGH);
    delay(10);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    distance = (duration/2) / 29.1;
    digitalWrite(voice, LOW);
    if (distance<40) {
        digitalWrite(voice, HIGH);
        delay(2000);
    }
    if (distance<20) {
        digitalWrite(voice, HIGH);
        delay(2000);
        delay(2000);
    }
    if (distance<10) {
        digitalWrite(voice, HIGH);
        delay(2000);
    }
}
```

## House

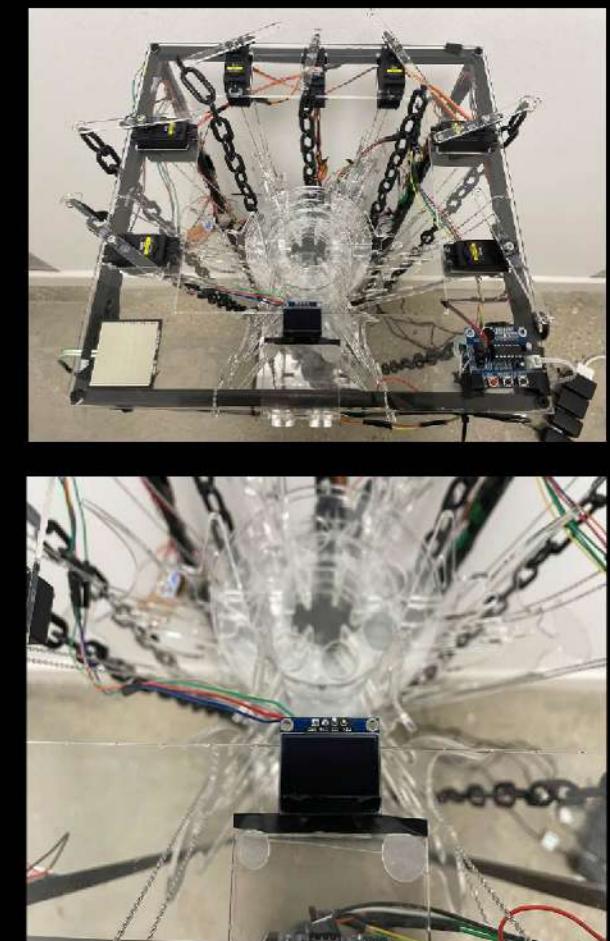
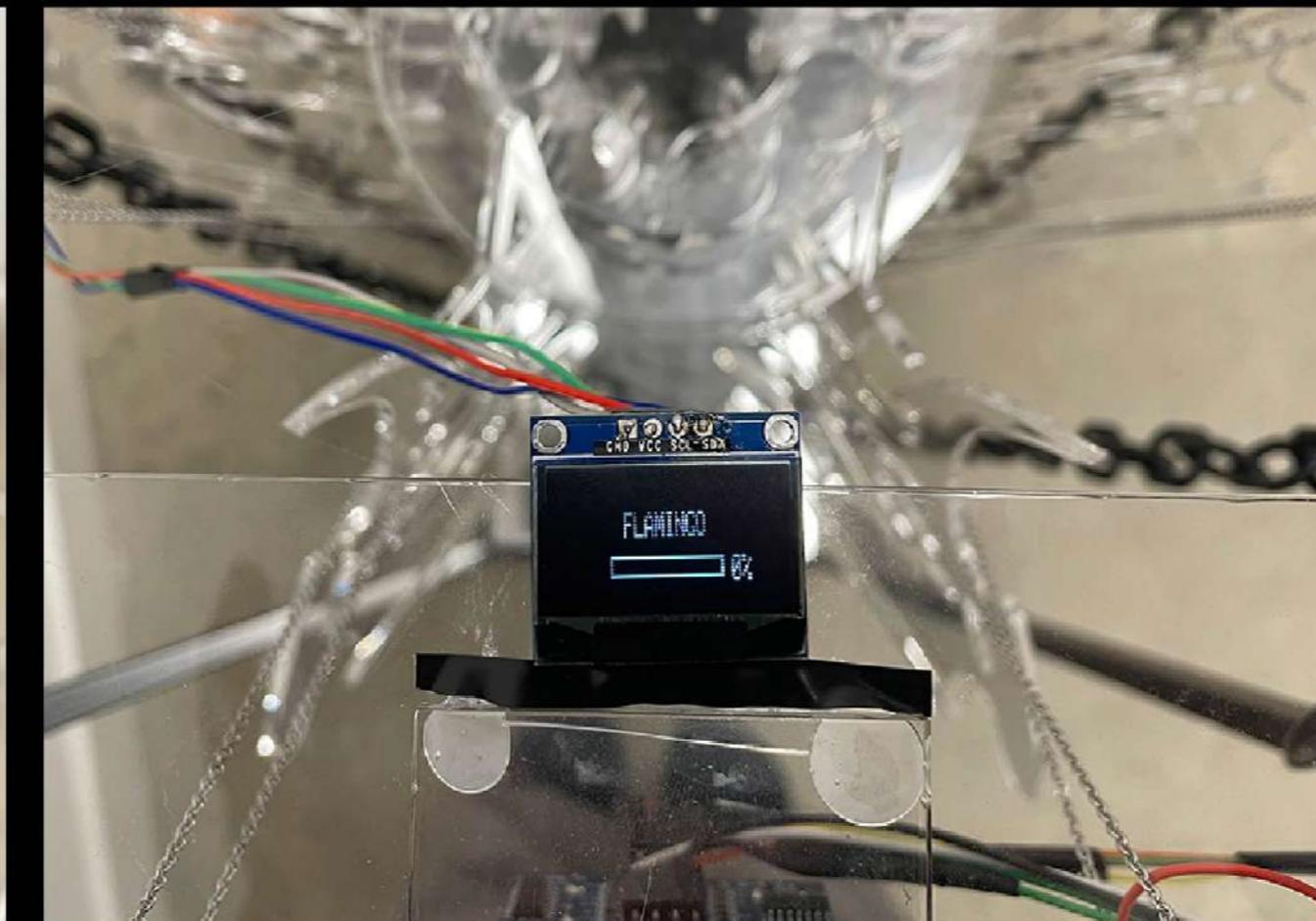
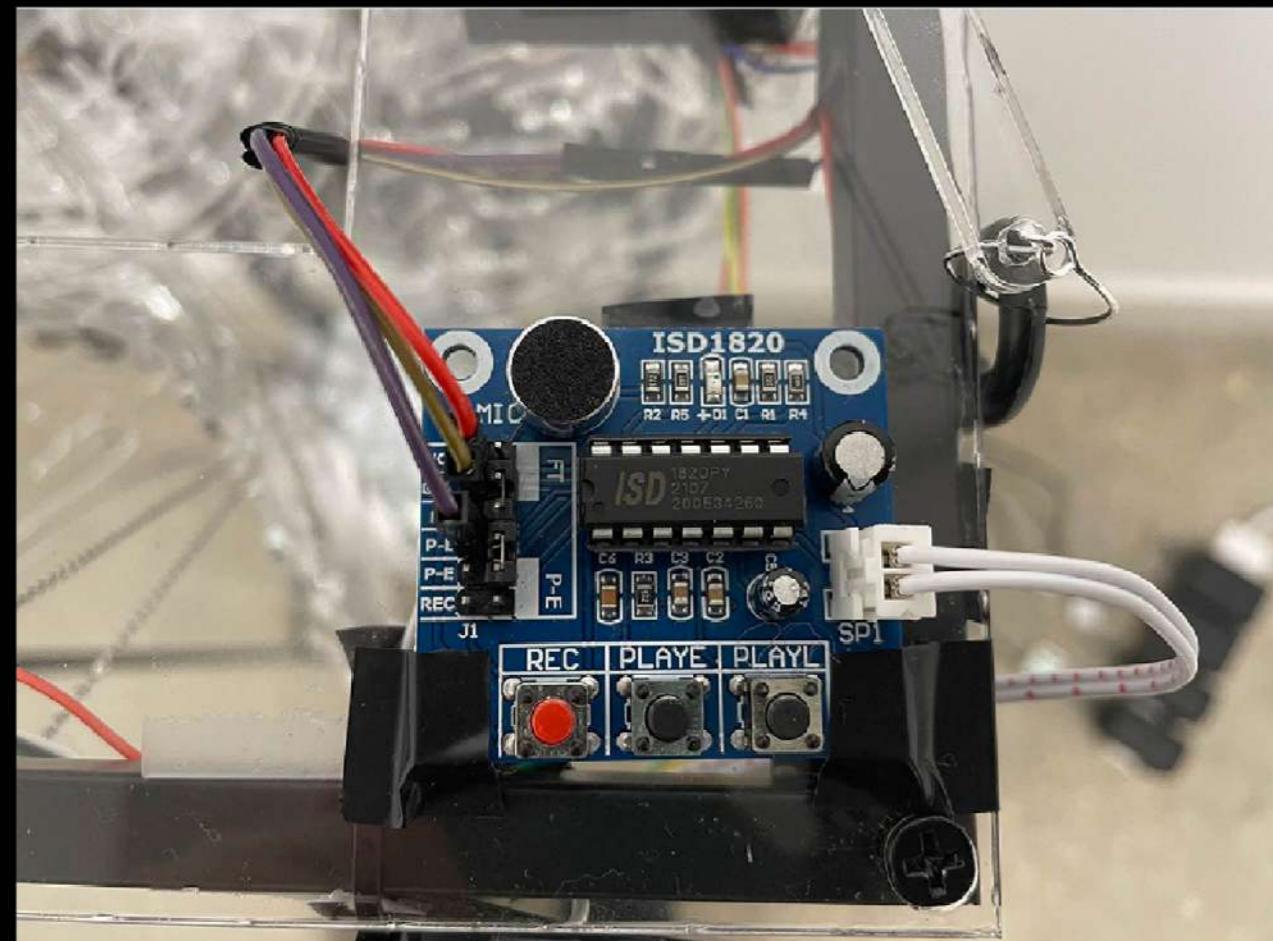
I had originally intended to laser cut a pattern of a hand then place the OLED screen in the “hand” and hang the “hand” over the installation. But in the end the visual effect didn't look very good. I had to place the OLED screen without any decoration on top of the installation.



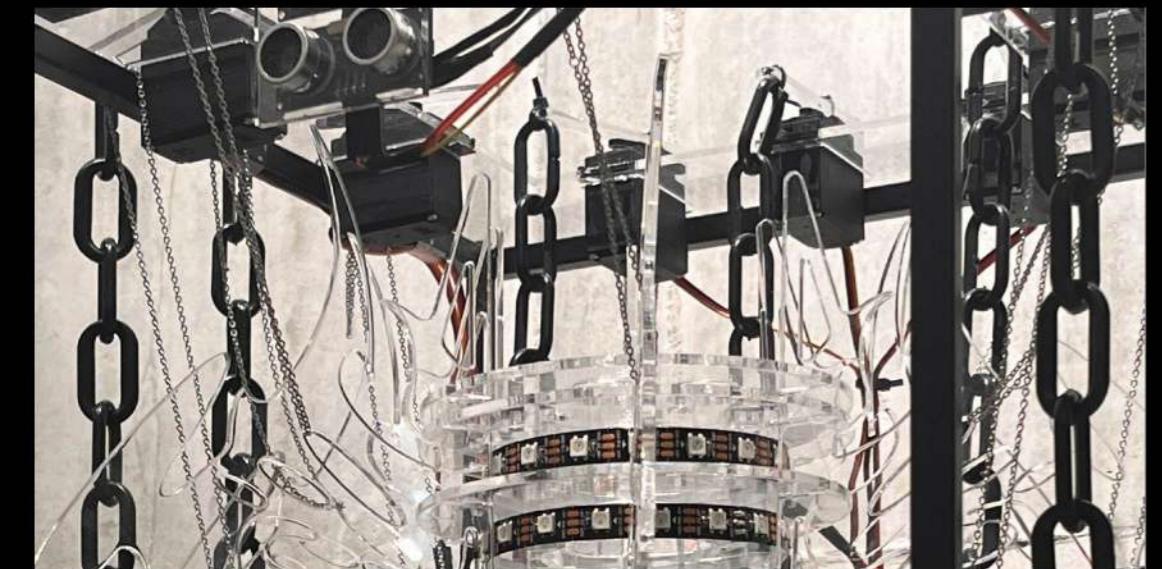
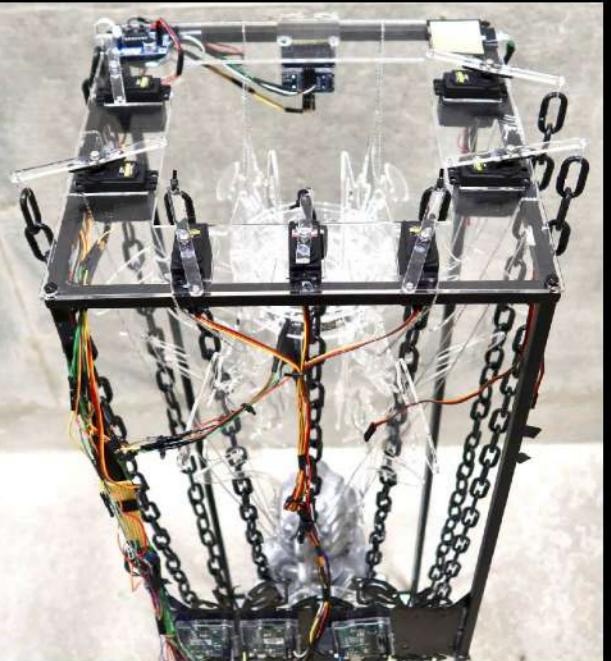
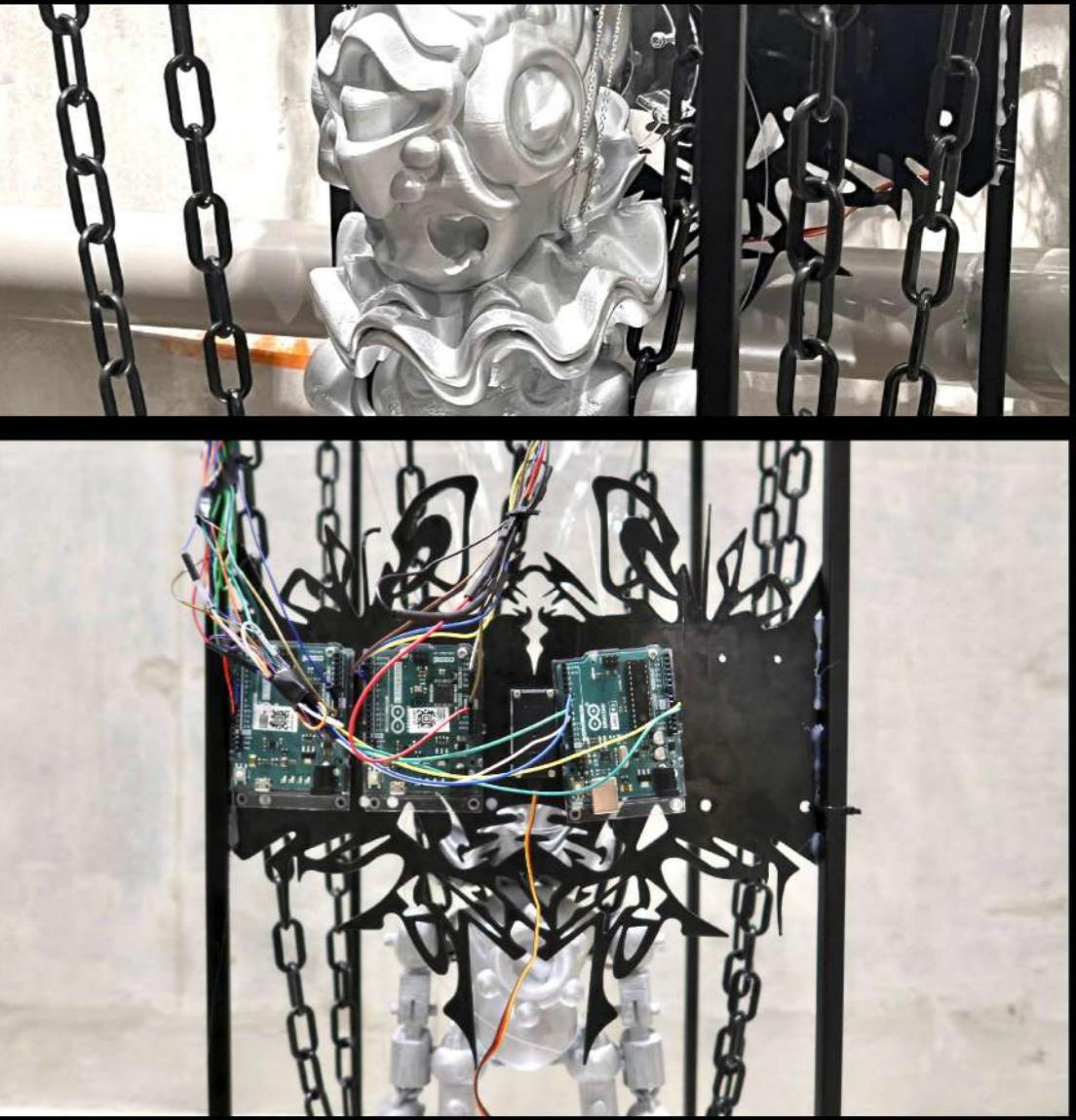
# Effect

## Sound

## OLED Screen



Final



# The code of using pressure sensor to control 4 servos

```
#include <Servo.h>

int fsr = 0;
int pos = 0;

Servo servo_1;
Servo servo_2;
Servo servo_3;
Servo servo_4;

void setup()
{
    pinMode(A0, INPUT);
    Serial.begin(9600);
    servo_1.attach(3, 500, 2500);
    servo_2.attach(4, 500, 2500);
    servo_3.attach(9, 500, 2500);
    servo_4.attach(6, 500, 2500);
}

void loop()
{
    fsr = analogRead(A0);

    if (fsr >= 0 && fsr < 50) {
        Serial.println(fsr); // getting know if the fsr go into this area
        for (pos = 0; pos <= 45; pos += 1) { // goes from 0 degrees to 45 degrees
            servo_1.write(pos); // tell servo to go to position in variable 'pos'
            servo_2.write(pos);
            servo_3.write(45+pos);
            servo_4.write(45+pos);
            delay(5); // waits 5 ms for the servo to reach the position
        }
        for (pos = 45; pos >= 0; pos -= 1) { // goes from 45 degrees to 0 degrees
            servo_1.write(pos); // tell servo to go to position in variable 'pos'
            servo_2.write(pos);
            servo_3.write(pos-45);
            servo_4.write(pos-45);
            delay(5); // waits 5 ms for the servo to reach the position
        }
        else if (fsr >= 50 && fsr <= 300){
            Serial.println(fsr);

            for (pos = 0; pos <= 90; pos += 1) { // goes from 0 degrees to 90 degrees
                servo_1.write(pos); // tell servo to go to position in variable 'pos'
                servo_2.write(pos);
                servo_3.write(pos+45);
                servo_4.write(pos+45);
                delay(5); // waits 5 ms for the servo to reach the position
            }
            for (pos = 90; pos >= 0; pos -= 1) { // goes from 90 degrees to 0 degrees
                servo_1.write(pos); // tell servo to go to position in variable 'pos'
                servo_2.write(pos);
                servo_3.write(pos-45);
                servo_4.write(pos-45);
                delay(5); // waits 5 ms for the servo to reach the position
            }
            else {
                Serial.println(fsr);
                for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
                    servo_1.write(pos); // tell servo to go to position in variable 'pos'
                    servo_2.write(pos);
                    servo_3.write(pos);
                    servo_4.write(pos);
                    delay(5); // waits 5 ms for the servo to reach the position
                }
                for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
                    servo_1.write(pos); // tell servo to go to position in variable 'pos'
                    servo_2.write(pos);
                    servo_3.write(pos+45);
                    servo_4.write(pos+45);
                    delay(5); // waits 5 ms for the servo to reach the position
                }
                //digitalWrite(HIGH);
            }
            //delay(10); // Delay a little bit to improve simulation performance
        }
    }
}
```

# The code of using light dependent sensor to control 3 servos

```
#include<Servo.h>
int lightval;
int lightpin=A0;
int pos = 0;
int tm=100;
int servo1pin=2;
int servo2pin=3;
int servo3pin=4;

Servo myservo1;
Servo myservo2;
Servo myservo3;

int angle;
void setup() {
    Serial.begin(9600);
    pinMode(lightpin,INPUT);
    myservo1.attach(servo1pin);
    myservo2.attach(servo2pin);
    myservo3.attach(servo3pin);
    pinMode(servo1pin,OUTPUT);
    pinMode(servo2pin,OUTPUT);
    pinMode(servo3pin,OUTPUT);
}

void loop() {
    lightval=analogRead(lightpin);
    Serial.println(lightval);
    delay(tm);

    angle= lightval/5;
    myservo1.write(angle);
    myservo2.write(angle);
    myservo3.write(angle);
    Serial.println("angle is");
    Serial.println(angle);

}
```

# The code of using leapmotion to control 1 servo

## Processing code

```
import com.onformative.leap.LeapMotionP5;
import com.leapmotion.leap.Finger;
LeapMotionP5 leap;

int angle;
import processing.serial.*;
Serial port;
public void setup() {
// set window, P3D = 3D rendering
size(720, 720, P3D);
noFill();
stroke(255);

// set LEAP object
leap = new LeapMotionP5(this);

// set com port. Currently: "/dev/tty.usbmodemfd121"
println("Available serial ports:");
println(Serial.list());
port = new Serial(this,Serial.list()[0], 9600);
}

public void draw() {
background(0);
fill(255);

for (Finger f : leap.getFingerList()) {
PVector position = leap.getTip(f);
//PVector velocity = leap.getVelocity(f);
ellipse(position.x, position.y, 10, 10);
if (position.y > 720) {
angle = 180;
} else if (position.y < 0) {
angle = 0;
} else {
angle = int((position.y) / 4);
}
port.write(angle);
println(angle);
}

public void stop() {
leap.stop();
}
```

## Arduino code

```
#include <Servo.h>
Servo myServo;
//int handPos;
//int angle;
void setup() {
myServo.attach(9);
Serial.begin(9600);
myServo.write(0);
}
void loop() {
byte angle;

if (Serial.available()) {
// Read angle from Processing
angle = Serial.read();
Serial.println(angle);
// If fingers in window, read servo angle
myServo.write(angle);
}
}
```

# The code of using ultrasonic sensor to control Neopixels

```
#include <Adafruit_NeoPixel.h>
#ifndef __AVR__
#include <avr/power.h> // Required for 16 MHz Adafruit Trinket
#endif

#define PIN      6
#define NUMPIXELS1 19
#define NUMPIXELS2 19

Adafruit_NeoPixel pixels1(NUMPIXELS1, PIN, NEO_GRB + NEO_KHZ800);
Adafruit_NeoPixel pixels2(NUMPIXELS2, PIN, NEO_GRB + NEO_KHZ800);

const int trigPin = 12;
const int echoPin = 11;
int duration = 0;
int distance = 0;

void NeoBlink(int num, int wait)
{
    if(distance <20)
    {Serial.print("00");
        for (int i = 0; i < num; i++)
        {
            pixels1.setPixelColor(i, 14, 221, 162);
            pixels2.setPixelColor(i, 14, 221, 162);
        }
        pixels1.show();
        pixels2.show();
    }
    else if(distance>20 && distance<40){
        Serial.print("22");
        for (int j = 0; j < num; j++)
        {
            pixels1.setPixelColor(j, 255, 77, 230);
            pixels2.setPixelColor(j, 255, 77, 230);
        }
        pixels1.show();
        pixels2.show();
        delay(10);
    }
    else {
        Serial.print("33");
        for (int p = 0; p < num; p++)
        {
            pixels1.setPixelColor(p, 217, 215, 226);
            pixels2.setPixelColor(p, 217, 215, 226);
            delay(10);
            for (int x = 255; x > 0; x=x-2){
                pixels1.setBrightness(x);
                pixels2.setBrightness(x);
                pixels1.show();
                pixels2.show();
                //delay(10);
            }
            pixels1.show();
            pixels2.show();
        }
        delay(wait);
    }
}

void setup()
{
    pinMode(trigPin , OUTPUT);
    pinMode(echoPin , INPUT);
    pixels1.begin();
    pixels2.begin();
    pixels1.setBrightness(50);
    pixels2.setBrightness(50);
    Serial.begin(9600);
}

void loop()
{
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    distance = duration/58.2;
    NeoBlink(19, 500);

    delay(100);
}
```

## OLED

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 32 // OLED display height, in pixels

#define OLED_RESET -1 // Reset pin # (or -1 if sharing
Arduino reset pin)
#define SCREEN_ADDRESS 0x3C ///< See datasheet for
Address; 0x3D for 128x64, 0x3C for 128x32
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT,
&Wire, OLED_RESET);

void setup() {
  Serial.begin(9600);

  // SSD1306_SWITCHCAPVCC = generate display voltage from
3.3V internally
  if(!display.begin(SSD1306_SWITCHCAPVCC,
SCREEN_ADDRESS)) {
    Serial.println(F("SSD1306 allocation failed"));
    for(); // Don't proceed, loop forever
  }
}

}
```

## Sound

```
#define trigPin 9
#define echoPin 8
#define voice 6
int sound = 250;
void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(voice, OUTPUT);
}
void loop() {
  Serial.begin(9600);
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delay(2);
  digitalWrite(trigPin, HIGH);
  delay(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;
  digitalWrite(voice, LOW);
  if (distance<40) {
    digitalWrite(voice, HIGH);
    delay(2000);
  }
  if (distance<20) {
    digitalWrite(voice, HIGH);
    delay(2000);
    delay(2000);
  }
  if (distance<10) {
    digitalWrite(voice, HIGH);
    delay(2000);
  }
}
```